

EXPRESS MAIL MAILING LABEL NUMBER EL52127 2657US

PATENT

10830.0080.NPUS00

**APPLICATION FOR UNITED STATES LETTERS PATENT**

for

**VIDEO FILE SERVER CACHE MANAGEMENT USING MOVIE RATINGS  
FOR RESERVATION OF MEMORY AND BANDWIDTH RESOURCES**

By

**Sorin Faibish  
John Forecast  
Ugur Sezer  
Peter Bixby  
Wayne W. Duso**

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates generally to video file servers and more particularly to a video file server providing "video on demand" (VOD) service. In particular, the present invention relates to management of cache memory and bandwidth resources in such a video file server.

### 2. Description of the Related Art

To provide "video on demand" (VOD) service, a multiplicity of digitally-encoded movies are stored in an array of disk drives, and the disk drives are coupled to a data network for delivery of the movies to network clients for viewing. It is desired to permit each client to select any one of the movies in the disk storage for viewing at any time, and to give the client "video cassette recorder" (VCR) functionality permitting the client to pause and fast-forward or fast-reverse to any location in the movie while viewing the movie.

Video file servers are typically constructed using disk drives, network interfaces, and other components that have been designed and manufactured for general data storage and network computing applications. For example, one architecture for a video file server includes a cross-bar switch to couple an array of conventional disk drives to an array of network interface circuits. In this architecture, it is possible for any network client communicating with any one of the network interface circuits to access any movie in any one of the disk drives. This architecture, however, has a disk drive bandwidth limitation preventing more than a certain number of independent real-time video streams to issue from any one of the disk drives. This bandwidth limitation is problematic for

1 providing video-on-demand service for the more popular movies. The bandwidth  
2 limitation can be overcome by storing a complete copy of each of the more popular  
3 movies in more than one of the disk drives, but such a solution is very costly. A more  
4 economical solution is to deny VCR functionality for the most popular movies, and  
5 instead offer the most popular movies starting at various increments of time and  
6 continuing without interruption. For example, if each disk drive has sufficient bandwidth  
7 to supply eight concurrent-time video streams, then it is possible for the one disk drive  
8 containing only one two-hour movie to permit any client to begin viewing at the start of  
9 the movie every fifteen minutes. Each of the eight video streams could be simulcast to  
10 any number of the clients that could be serviced concurrently by all of the network  
11 interface circuits.

12 Another architecture for a video file server is a hierarchical architecture that uses  
13 a cached disk storage system and network interface circuits in a multiplicity of  
14 commodity computers. The cached disk storage system includes a primary cache  
15 memory, and each commodity computer includes a local secondary cache memory. Each  
16 cache memory provides a much higher available bandwidth than a single disk drive.  
17 Therefore, it is possible to store an entire movie in cache memory to provide the network  
18 clients with VOD and VCR access to the movie. An example of such a hierarchical  
19 video file server is described in Duso et al. U.S. Patent 5,892,915 issued April 6, 1999,  
20 incorporated herein by reference. Such a video file server is manufactured and sold by  
21 EMC Corporation, 35 Parkwood Drive, Hopkinton, MA 01748.

22 Although Duso et al. U.S. Patent 5,892,915 teaches how to make a video file  
23 server that can provide VOD and VCR access to the most popular movies and also

1 unpopular movies, there is a need for more efficient management of access to popular  
2 movies that are not sufficiently popular to always keep them in cache memory.  
3 Moreover, the popularity of a movie will change with time, and therefore it is desired for  
4 the video file server to manage cache memory and bandwidth in accordance with the  
5 changing popularity of the movie.

## 7 SUMMARY OF THE INVENTION

8 In accordance with one aspect, the present invention provides a video file server  
9 for providing clients with video-on-demand access to movies. The video file server  
10 includes a cached disk storage system including a primary cache memory and disk  
11 storage for storing the movies; and a multiplicity of data mover computers coupled to the  
12 cached disk storage system for streaming video data from the cached disk storage system  
13 to clients in a data network. Each of the data mover computers has a local cache  
14 memory. The movies are ranked with respect to popularity, and a respective set of the  
15 data movers are pre-assigned for servicing video streams for each movie ranking.

16 In accordance with another aspect, the invention provides a video file server for  
17 providing clients with video-on-demand access to movies. The video file server  
18 comprises a cached disk storage system including a cache and disk storage for storing the  
19 movies; and a multiplicity of data mover computers coupled to the cached disk storage  
20 system for streaming video data from the cached disk storage system to clients in a data  
21 network. The video file server is programmed for locking in the cache a plurality of  
22 entire movies, and when there is a need for servicing a more popular movie from the  
23 cache and there is insufficient free cache memory for servicing the more popular movie

1 from the cache, transferring the servicing of a less popular movie from the cache to disk  
2 storage in order to free cache memory for servicing the more popular movie from the  
3 cache.

4 In accordance with yet another aspect, the invention provides a method of  
5 operating a video file server for providing clients with video-on-demand access to  
6 movies. The video file server has a cached disk storage system including a primary cache  
7 and disk storage containing the movies, and a multiplicity of data mover computers  
8 coupled to the cached disk storage system for streaming video data from the cached disk  
9 storage system to clients in a data network. Each of the data mover computers has a local  
10 cache. The method includes ranking the movies with respect to popularity, and assigning  
11 a respective set of the data movers to each movie ranking, and servicing video streams for  
12 each movie ranking with the respective set of data movers assigned for servicing said  
13 video streams for said each movie ranking.

14 In accordance with still another aspect, the invention provides a method of  
15 operating a video file server for providing clients with video-on-demand access to  
16 movies. The video file server has a cached disk storage system including a cache and  
17 disk storage containing the movies, and a multiplicity of data mover computers coupled  
18 to the cached disk storage system for streaming video data from the cached disk storage  
19 system to clients in a client data network. The method includes locking in the cache a  
20 plurality of entire movies, and when there is a need for servicing a more popular movie  
21 from the cache and there is insufficient free cache memory for servicing the more popular  
22 movie from the cache, transferring the servicing of a less popular movie from the cache

1 to the disk storage in order to free cache memory for servicing the more popular movie  
2 from the cache.

#### 3 4 **BRIEF DESCRIPTION OF THE DRAWINGS**

5 Other objects and advantages of the invention will become apparent upon reading  
6 the following detailed description with reference to the drawings, in which:

7 FIG. 1 is a block diagram of a hierarchical video file server incorporating the  
8 present invention;

9 FIG. 2 is a graph of frequency of access in plays per day versus the ranking of  
10 movies stored in the video file server of FIG. 1;

11 FIG. 3 is a table showing a correspondence between the movie ranking and a data  
12 mover set aside for servicing client requests for access to the movie;

13 FIG. 4 shows an example of how the data movers are physically configured  
14 differently for providing client access to movies of respective ranks;

15 FIG. 5 shows an example of a portion of the service database in the control station  
16 associating certain information with the movies;

17 FIG. 6 shows an allocation of reserved and not reserved streams, and active and  
18 inactive streams from a data mover set;

19 FIG. 7 shows an allocation of bandwidth on the storage system bus;

20 FIG. 8 shows an allocation of primary cache memory capacity;

21 FIG. 9 shows a number of entire movies locked in the primary cache;

22 FIG. 10 shows various ways of indicating to a client the availability of movie  
23 titles during a high demand situation;

FIG. 11 is a flowchart showing how movie availability is indicated to a client; and  
FIGs. 12 to 20 comprise pages of a flowchart showing how the video file system  
of FIG. 1 is programmed to respond to a client's request for video service.

While the invention is susceptible to various modifications and alternative forms,  
specific embodiments thereof have been shown by way of example in the drawings and  
will be described in detail. It should be understood, however, that it is not intended to  
limit the invention to the particular forms disclosed, but on the contrary, the intention is  
to cover all modifications, equivalents, and alternatives falling within the scope of the  
invention as defined by the appended claims.

## DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

With reference to FIG. 1, of the drawings, there is shown a video file server  
generally designated 20 incorporating the present invention. The video file server 20  
includes an array of data mover computers 21, at least one control station computer 22,  
and a cached disk storage system 23. The video file server 20 is a high performance,  
high capacity, and high-availability network-attached data server. It provides the ability  
for multiple file systems to exist concurrently over multiple communication stacks, with  
shared data access. It also allows multiple physical file systems to co-exist, each  
optimized to the needs of a particular data service.

The video file server 20 is managed as a dedicated network appliance, integrated  
with popular network operating systems in a way, which, other than its superior  
performance, is transparent to the end user. It provides specialized support for  
isochronous data streams used in live, as well as store-and forward, audio-visual

1 applications. Therefore, the video file server 20 is suitable for a wide variety of  
2 applications such as image repositories, video on demand, and networked video  
3 applications, in addition to high-end file server applications.

4 The cached disk storage system 23 includes disk storage 24 in the form of an  
5 array of small commodity disk drives. The cached disk storage system 23 also includes a  
6 high-capacity random access cache memory 25. The disk storage 24 and the cache  
7 memory 25 share a system bus 26. The cached disk storage system 23 also includes a  
8 number of port adapters 27, and each of the data mover computers 21 and the control  
9 station computer 22 is linked to the system bus 26 by at least one of the port adapters.

10 Each of the data movers 21 is a high-end commodity computer. The clustering of  
11 the data movers 21 as a front end to the cached disk storage system 23 provides  
12 parallelism and scalability. Each of the data mover computers 21 includes a respective  
13 local cache 28, 29, 30 of random access memory. The local caches function as a very  
14 large capacity secondary cache, and the cache 25 in the cached disk storage system 23  
15 functions as a primary cache, in a hierarchical storage architecture.

16 For real-time streaming of movie data, the storage and access of the movie data in  
17 the cache memory is scheduled by the control station 22 in accordance with an admission  
18 control protocol. The control station 22 is linked to the data movers 21 by a dual-  
19 redundant high-speed internal Ethernet network 31. The control station 22 responds to a  
20 client request for the real-time streaming of movie data by checking whether or not the  
21 video file server has sufficient memory and bandwidth resources to guarantee that when a  
22 request is granted, the real-time streaming can continue without interruption until the end  
23 of the movie. The resources currently in use and the resources available are noted in a



1 service data base 32 maintained in local memory of the control station 22 and backed-up  
2 in the disk storage 24. If there are sufficient resources, then the control station 22 places  
3 locks on the required resources, grants the client's request, and maintains the locks until  
4 the client's request is satisfied. For example, the control station 22 may determine that a  
5 video stream can be serviced by pre-fetching the movie data from the disk storage 24 and  
6 buffering the data through the primary cache 25 and one of the data mover local caches  
7 28, 29, 30. In this case, the control station 22 will select one of the data movers to  
8 service the client request, and will place a lock upon certain bandwidth and cache  
9 memory resources of the selected data mover, and will also place a lock on bandwidth  
10 resources of the system bus 26 and memory resources of the primary cache 25, and these  
11 locks will remain in place until the selected data mover is finished streaming video data  
12 to the client.

13 Further details regarding the construction and operation of a video file server are  
14 found in Duso et al. U.S. Patent 5,892,915 issued April 6, 1999, incorporated herein by  
15 reference. For practicing the present invention, the video file server 20 is programmed as  
16 further described below. In addition, the physical configuration of the data movers may  
17 also be modified as described below with reference to FIG. 4. These modifications  
18 provide more efficient management of access to popular movies that are not sufficiently  
19 popular to always keep them in cache memory.

20 Proper management of cache memory in the video file server of FIG. 1 is  
21 important for effective use of the bandwidth resources in the video file server of FIG. 1.  
22 FIG. 1 has been drawn to emphasize the hierarchical architecture of the file server. At  
23 the top of the hierarchy, at the disk storage level, the bandwidth resources are at a

1 minimum. At the bottom of the hierarchy, at the data mover interface to the client data  
2 network 33, the bandwidth resources are at a maximum. The increased bandwidth  
3 resources at the bottom of the hierarchy can be fully used only if the same data from the  
4 disk storage 24 at the top of the hierarchy is being streamed to multiple clients at the  
5 bottom of the hierarchy. Unless the same data is being multicast from the disk storage 24  
6 to the multiple clients, it must be streamed from cache memory, and there must be a  
7 balance between the increased bandwidth resources at the bottom of the hierarchy and the  
8 cache memory capacity of the primary cache 25 and the data mover local cache memories  
9 28, 29, and 30.

10 For a few of the most popular movies, an entire movie can be kept in the local  
11 cache of one or more of the data movers 21. For unpopular movies, there is no advantage  
12 to keeping an entire movie in cache because it is most likely that the limited number of  
13 clients simultaneously viewing the movie can be supported by streaming the video data  
14 directly from one disk drive containing the movie. Between these two extremes,  
15 however, there are a substantial number of rather popular movies that should be serviced  
16 from cache in an appropriate fashion to satisfy client demands for VOD and VCR  
17 service.

18 It is possible to configure the hardware of the video file server 20 for effective  
19 service of all kinds of movies due to a certain regularity in the frequency of access of  
20 movies once the movies have been ranked in terms of their frequency of access. Shown  
21 in FIG. 2, for example, is an idealized plot of the frequency of access, in plays per day,  
22 the ranking of each movie accessible from the video file server. The most popular movie  
23 is given a rank of one, the next most popular movie is given a rank of 2, and so on. When

1 plotted on log-log paper, the plot falls on a straight line with negative slope in accordance  
2 with the "Zipf" distribution. (See, for example, the Jakob Nielsen, "Zipf Curves and  
3 Website Popularity," published at <http://www.useit.com/alertbox/zipf.html>). Although the  
4 popularity of each movie may rise and will fall with time, the plot will retain its general  
5 characteristic of a line having a certain slope and intercept with the "y" axis. Therefore,  
6 it is possible to configure and program the video file server in accordance with the  
7 general characteristic, in order to handle effectively each movie ranking. Moreover, the  
8 video file server can be configured and programmed initially without regard to the rank  
9 of any particular movie.

10 When a movie is first released for VOD distribution, it will have an industry  
11 rating, such as the "Blockbuster" rating, with respect to other popular movies. Therefore,  
12 the newly released movie can be ranked with respect to the popular movies already stored  
13 in the video file server, and the rank of each lower rated movie in the video file server  
14 can be decreased by one level as the newly released movie is written to disk storage of  
15 the video file server. New movie releases, for example, are written into the disk storage  
16 at the time of day of minimum demand, for example, about 3:00 a.m. Each day, the  
17 number of accesses for each movie is recorded to compute a running average of the  
18 frequency of access of each movie, in order to re-adjust the rank of each movie, and make  
19 effective the adjusted rank at the time of day of minimum demand.

20 In accordance with one aspect of the present invention, for each movie ranking, a  
21 particular set of data mover resources are preassigned for servicing client access to the  
22 movie having the movie ranking. With reference to FIG. 3, for example, there is shown a  
23 table associating each movie ranking with a data mover set pre-assigned to service the

1 movie having the associated ranking. Therefore, there is a certain maximum number of  
2 video streams that are available to each movie, corresponding to the total number of  
3 video streams that all of the data movers in the set can supply to the client data network.  
4 There is also a certain maximum amount of data mover local cache memory capacity  
5 available for servicing each movie, corresponding to the total amount of local cache  
6 memory in all of the data movers in the set.

7 For the highest ranked movie (i.e., the most popular movie having the lowest  
8 numerical rank value), the data mover set may include a multiplicity of data movers, in  
9 order to support a large number of concurrent video streams. For the lowest ranked  
10 movies (i.e., unpopular movies), the same data mover set consisting of a single data  
11 mover may service a multiplicity of movies. The data mover sets need not be disjoint,  
12 and instead there may be overlap between the data mover set servicing one movie ranking  
13 and the data mover set servicing the movie ranking having the next higher or lower rank.  
14 The overlap may facilitate the process of making effective the adjustment in rank  
15 performed at the time of day of minimum demand, because in some instances there will  
16 be no change in at least one of the data movers assigned to service a movie having its  
17 rank changed to the next higher or lower rank. For example, when there is a change in  
18 rank of a movie so that a movie previously serviced by a first data mover will become  
19 serviced by a second data mover, there may be a movement of the movie from the local  
20 data mover cache of the first data mover to the local data mover cache of the second data  
21 mover. This process should be transparent to a client currently viewing the movie.  
22 However, if there are a large number of clients currently viewing the movie, the VCR  
23 functionality may be suspended for a brief moment during the time for the transfer of the

1 movie between the caches, so that the video stream for each client currently viewing the  
2 movie is switched only once from the first data mover to the second data mover.

3 In many cases where the transfer of the movie occurs between a first data mover  
4 cache to a second data mover cache, the association of each movie ranking with a  
5 respective set of data movers offers the advantage that the transfer can occur over a  
6 dedicated link between the two data movers. For example, as shown in FIG. 1, the data  
7 movers are arranged in a series such that neighboring data movers in the series are  
8 coupled by a dedicated link 34, 35, etc. and the flow of movie data from one data mover  
9 to the next over a dedicated link permits the movie data from one data mover set  
10 servicing one movie ranking to be directly transferred to another data mover set servicing  
11 a next higher or lower movie ranking. In many cases the transfer will occur in this  
12 fashion because in the time span of one day the data mover set association of a movie is  
13 likely to change if at all from one data mover set to servicing one movie ranking to  
14 another data mover set servicing a next higher or lower ranking.

15 As shown in FIG. 4, the physical configuration of each of the data movers 21  
16 depends on the rank of the movie or movies to be serviced by the data mover. In  
17 particular, the data movers servicing the higher ranking (i.e., the most popular) movies  
18 have fewer cache RAM cards 41 and more network interface cards 42 than the data  
19 movers servicing the lower ranking (i.e., less popular) movies. This is a consequence of  
20 the fact that the cache RAM has sufficiently high bandwidth that a large number of video  
21 streams can be serviced from one copy of a movie in the cache RAM. For less popular  
22 movies, not all of this bandwidth can be utilized, so that the ratio of network interface  
23 cards to cache RAM cards will fall for less popular movies.

1 For the most popular movies, there can be multiple data movers servicing just one  
2 movie, and in this case only a fraction of the cache RAM for servicing an entire movie  
3 need reside in any one of the movies. For example, as shown in FIG. 4, the movie data  
4 for movie #1 could be circulated among the first three data movers, and the movie data  
5 for movie #2 could be circulated among the third and fourth data movers. The data  
6 mover #1 could have a cache capacity to hold 40% of movie #1, the data mover #2 could  
7 have a cache capacity to hold 40% of movie #1, the data mover #3 could have a cache  
8 capacity to hold 25% of movie #1 and 40%% of movie #2, and the data mover #4 could  
9 have a cache capacity to hold 65% of movie #2. Data mover #6 could have a cache  
10 capacity to hold at least 100% of movie #4.

11 FIG. 5 shows an example of a portion of the service database in the control station  
12 associating certain information with the movies. For each movie, the service database  
13 includes the movie ranking, the movie title, the frequency of access (in accesses per day),  
14 the data mover set for servicing client access to the movie, a certain number of streams  
15 reserved for accessing the movie, the number of active video streams from the movie, the  
16 location of the movie on disk, and any location of the movie in cache.

17 FIG. 6 shows an allocation of reserved and not reserved streams, and active and  
18 inactive streams from a data mover set. The programming of the video file server takes  
19 into account the fact that there may be certain times of day, such as 8 p.m. to 11 p.m.,  
20 when the server cannot satisfy all requests for VOD service. This will be referred to as a  
21 peak demand situation. During the peak demand situation, VOD service is likely to be  
22 denied for the less popular movies, unless a less popular movie already happens to be in  
23 cache and being streamed to a fair number of clients. The reservation of a certain number

1 of streams for servicing each movie prevents an allocation of streams prior to a high  
2 demand condition from becoming frozen during the high demand period. This can be a  
3 significant advantage if the demand for certain movies (such as adult films) has a  
4 tendency to rise, and the demand for certain other movies (such as family films) has a  
5 tendency to fall, during the peak demand period.

6 FIG. 7 shows an allocation of bandwidth on the storage system bus. The  
7 bandwidth is allocated for access of movies from the primary cache, or for access of  
8 movies from disk. The difference between the total available bus bandwidth and the  
9 allocated bandwidth is the free bandwidth.

10 FIG. 8 shows an allocation of primary cache memory capacity. A certain amount  
11 of the cache memory can be locked for access of one or more entire movies in the cache.  
12 The cache memory can also be locked for buffering of disk access. The difference  
13 between the total primary cache memory and the allocated cache memory is the free  
14 cache memory.

15 FIG. 9 shows a number of entire movies locked in the primary cache. Associated  
16 with each movie locked in the primary cache is a certain number of streams serviced  
17 from the movie in the cache. If the number of streams serviced from an entire movie in  
18 the primary cache falls below the number that can be serviced from disk, then it may be  
19 advantageous to service the streams from disk in order to free up cache memory for  
20 servicing a more popular movie from cache.

21 FIG. 10 shows various ways of indicating to a client the availability of movie  
22 titles during a high demand situation. The client may request the availability of a  
23 specified movie title. The client may also browse through a series of listing of groups of

1 movies having varying degrees of availability. The first group includes the movies that  
2 are accessible to start at any time with full VCR functionality. The next series of groups  
3 are movies that are multicast from disk without VCR capability, starting at periodic  
4 increments of time, such as in the next 15 minutes, 30 minutes, etc. Then there is a group  
5 of movies that are already playing, from which could be spawned an additional stream to  
6 the client. For example, when the high-demand situation is reached, VCR functionality  
7 for popular movies not in cache is suspended in order to create the multi-cast streams  
8 spaced at periodic intervals in time.

9 FIG. 11 is a flowchart showing how movie availability is indicated to a client. In  
10 a first step 51, execution continues to step 52 if resources are not available to start  
11 immediately a new stream from any movie in disk storage. In step 52, the client is told  
12 that the system is at peak demand, and full service is expected at a particular time in the  
13 future. In step 53, the client is shown a list of titles of movies in cache that are accessible  
14 to start at any time. In step 54, the client is shown a list of titles of movies that are  
15 scheduled to begin at certain times. In step 55, the client is shown a list of titles of  
16 movies that are not accessible at any time but are currently playing. In step 56, the client  
17 is told that the following movies are available during off-peak hours. Then in step 58, the  
18 client is shown a list of titles of movies in disk storage. In step 59, the client is told that  
19 the following movies can be ordered from archival storage. In step 60, the client is  
20 shown a list of titles of movies in archival storage.

21 In step 51, if resources are available to start any movie in disk storage, then  
22 execution branches from step 51 to step 57. In step 57, the client is told that the  
23 following movies may start at any time. Execution then continues to step 58.



FIGs. 12 to 20 comprise pages of a flowchart showing how the video file system of FIG. 1 is programmed to respond to a client's request for video service. This request is received by the control station (22 in FIG. 1). In a first step 71, the control station (based on initial state in the service data base 32 in FIG. 1) selects a data mover DMj to serve the client. The control station sends a message over the high-speed serial bus (31 in FIG. 1) requesting DMj to respond to the client request. Then in step 72, the data mover DMj connects to the client. In step 73, the client negotiates a movie title with the data mover DMj, for example, as described above with reference to FIGS. 10 and 11, so that the data mover DMj obtains from the client a title of a movie that has been placed in disk storage. In step 74, the data mover DMj checks whether it has access to the movie. For example, DMj must be in the set of data movers assigned to service the rank of the movie. It is also possible that DMj is in the set of data movers assigned to service the rank of the movie but the movie should be serviced by another one of the data movers in the set, for example, because the portion of the movie to be initially accessed presently resides in the local cache of another one of the data movers in the set. If the titled movie is not accessible by the data mover DMj, then execution branches to step 75. In step 75, DMj informs the control station (CS) of the movie title. Then in step 76, the control station selects a data mover having access to the titled movie, and this selected data mover will be referenced as DMj in subsequent steps. Execution then continues to FIG. 13. Execution also continues to FIG. 13 from step 74 if the title is accessible by DMj.

In FIG. 13, in step 81, execution branches depending on whether or not DMj is already servicing a stream from the titled movie Mi. If not, execution branches to step 82. In step 82, execution branches depending on whether or not the titled movie Mi is in

1 the primary cache already. If not, then execution continues to step 83. In step 83,  
2 execution branches depending on whether or not there is cache available for the entire  
3 movie Mi. If not, execution continues to FIG. 19. Otherwise, execution branches to step  
4 86. In step 86, execution branches to FIG. 17 if primary cache bandwidth is available for  
5 servicing the movie from the primary cache. If so, execution continues to FIG. 17. If  
6 not, execution branches to step 87 to inform the control station of the lack of primary  
7 cache bandwidth.

8 In step 81, if the data mover DMj is already servicing a stream from the movie,  
9 then execution continues to step 84. Execution also continues to step 84 from step 82 if  
10 the title Mi is already in the primary cache. In step 84, the required bandwidth for a new  
11 stream (BW<sub>i</sub>) is computed, and for comparison, in step 85 the available (i.e., free)  
12 primary cache bandwidth (BW) is obtained. Execution then continues in step 86.

13 With reference to FIG. 14, in step 91, if disk bandwidth is not available, execution  
14 branches to FIG. 15. Otherwise, execution continues to step 92. In step 92, the control  
15 station opens the file for the movie Mi and sends a pointer to the file to the data mover  
16 DMj. Then in step 93, the control station logs in the service data base (DB) the fact that a  
17 stream for the movie Mi is being serviced from disk. In step 94, the data mover DMj  
18 starts service and continues service, until in step 95 the service time has expired. Once  
19 the service time has expired, execution continues to FIG. 16.

20 With reference to FIG. 15, in step 101, if the movie Mi is not in the data mover  
21 DMj local cache, execution continues to step 102. In step 102, if the movie Mi is in the  
22 local cache of any other data mover DMk, then execution continues to step 103. In step  
23 103, the data mover DMj requests access to the local cache of the data mover DMk. In

1 step 104, if the request is granted, execution continues to step 105, to start transfer of the  
2 movie Mi from the data mover DMk to the data mover DMj. Execution continues from  
3 step 105 to FIG. 17.

4 In step 104, if the request for access to the local cache of the data mover DMk is  
5 not granted, execution continues to step 106. In step 106, execution loops back to step  
6 103 if there is another data mover DMk in the data set for the ranking of the movie Mi.

7 In step 101, if the movie Mi is in the local cache of the data mover DMj, then  
8 execution branches to step 107. In step 107, if the data mover DMj has available  
9 bandwidth (BW) and processor (CPU) resources to service the movie, then execution  
10 continues to FIG. 17. Otherwise, execution continues to step 108. Execution also  
11 continues to step 108 from step 102 if the movie Mi is not in the local cache of any other  
12 data mover, and to step 108 from step 106 if no other data mover DMk is available for  
13 servicing the movie. In step 108, the data mover denies service by informing the control  
14 station (CS) that service cannot be provided. In step 109, the control station (CS) warns  
15 an operator of the denial of service condition, and updates a model of the availability of  
16 service in the service database (DB). Then in step 110, the control station returns a denial  
17 of service message to the user. Then in step 111, the control station logs the denial of  
18 service event. The log, for example, is periodically inspected to adjust the allocation of  
19 data movers to each movie ranking in order to improve overall service, or to determine  
20 what additional resources would be needed to improve service.

21 With reference to FIG. 16, in step 121, the data mover DMj informs the control  
22 station (CS) of the end of service. In step 122, the control station updates the service  
23 database (DB) so that the service to the client is no longer indicated. In step 123, the

1 number of services (ti) for the movie Mi from the primary cache is decremented by one.  
2 In step 124, the cache supporting the stream is released or freed. In step 125, the  
3 bandwidth of the cache and the bandwidth of the disk are updated to indicate that the  
4 bandwidth no longer needed for servicing the stream has become free. In step 126, the  
5 data mover (DM) prepares for new service. In step 127, if the number of streams (ti)  
6 serviced from the entire movie Mi in the primary cache is not zero, then execution loops  
7 back to FIG. 12 to service another client request. Otherwise, if the number of streams (ti)  
8 is zero, then execution branches from step 127 to step 128 to free the locked primary  
9 cache allocated to the entire movie Mi and to update the service data base (DB) to  
10 indicate that the movie Mi is no longer in the primary cache. Execution continues from  
11 step 128 to FIG. 12 to service another client request.

12 With reference to FIG. 17, in step 131, execution continues to step 132 if the  
13 entire movie Mi is not locked in the primary cache. In step 132, execution continues to  
14 step 133 if the data mover DMj is servicing only the movie Mi. In step 133, if the movie  
15 Mi is not in the local cache of the data mover DMj, then execution continues to FIG. 18.

16 In step 132, if the data mover DMj is serving only the movie Mi, then execution  
17 continues to step 134. In step 134, if the movie Mi is not in the local cache of the data  
18 mover DMj, then execution continues to FIG. 18. Otherwise, execution continues from  
19 step 134 to step 135. Execution also branches from step 133 to step 135 when the movie  
20 Mi is in the local cache of the data mover DMj. In step 135, preparations are made to  
21 service the movie Mi from the local cache of the data mover DMj, and the data mover  
22 DMj starts the stream in step 136. Execution continues from step 136 to FIG. 14.

1 In step 131, if the entire movie  $M_i$  is in locked primary cache, then execution  
2 branches to step 137. In step 137, the number of streams ( $t_i$ ) of the movie  $M_i$  serviced  
3 from the primary cache is incremented by one. Execution continues from step 137 to step  
4 136.

5 With reference to FIG. 18, in step 141, the data mover  $DM_j$  moves the movie  $M_i$   
6 from disk to primary cache by sending a pre-fetch command to the cached disk storage  
7 system. In step 142, the number of streams ( $t_i$ ) for the movie  $M_i$  is set to one. In step  
8 143, the data mover  $DM_j$  sends data to the control station (CS) for updating the service  
9 data base (DB) to indicate that the primary cache has been locked for storage of the  
10 movie  $M_i$  in the primary cache. In step 144, the data mover  $DM_j$  opens the file on the  
11 primary cache, to begin reading movie data from the primary cache. In step 145,  
12 execution branches to step 146 if the movie  $M_i$  is the first service of the data mover  $DM_j$ .  
13 In this case, there will be sufficient free local cache memory in the data mover  $DM_j$  to  
14 store the entire movie  $M_i$ . In step 146, the data mover  $DM_j$  starts caching the movie  $M_i$   
15 in its local cache. Execution continues from step 146 to step 147. Execution also  
16 continues to step 147 from step 145 when  $M_i$  is not the first service of the data mover  
17  $DM_j$ . In step 147, the data mover  $DM_j$  starts service of a video stream from the movie  
18  $M_i$ , and execution continues to FIG. 14.

19 With reference to FIG. 19, in step 151, if cache bandwidth (BW) is not available,  
20 execution branches to FIG. 14. If cache bandwidth (BW) is available, execution  
21 continues to step 152. In step 152, the service data base (DB) is read to obtain the  
22 number of services or streams  $t_i$  of the movies  $M_i$  in the primary cache and the respective  
23 rankings of these movies in cache. From these rankings, in step 153, the ranking of the

1 lowest ranking (i.e., the least popular) movie is found that is already in the primary cache.  
2 In step 154, the ranking of the requested movie  $M_i$  is obtained, for example, from a fixed  
3 one of the data movers assigned to service the movie. In step 155, if the ranking of the  
4 requested movie is not greater than the ranking of the lowest ranking movie in the  
5 primary cache (i.e., the requested movie is less popular than the least popular movie in  
6 the primary cache), then execution branches to FIG. 14. Otherwise, execution continues  
7 to FIG. 20.

8 With reference to FIG. 20, in step 161, if the number of streams ( $tk$ ) for the lowest  
9 ranking (i.e., least popular) movie in the primary cache is greater than a threshold, then  
10 execution continues to FIG. 14. In this case, it would be too disruptive to move the  
11 origination of the streams from an entire movie in cache to the movie on disk. Otherwise,  
12 execution continues from step 161 to step 162. In step 162, execution branches to FIG.  
13 14 if there is not enough bandwidth to remove the movie from locked primary cache and  
14 continue access of the movie from disk. This is done by comparing the sum of the bit  
15 rates for all of the streams from the lowest ranking (i.e., least popular) movie in the  
16 primary cache to the difference between the free disk bandwidth less the bit rate for the  
17 new stream. Otherwise, if there is enough bandwidth, execution continues to step 163.  
18 In step 163, the control station (CS) opens the file from disk for the movie  $M_k$  that was in  
19 the locked primary cache and is to be accessed from disk. In step 164, the control station  
20 (CS) sends a pointer to the data movers in the data mover set (DMs) servicing the movie  
21  $M_k$ . In step 165, the movie  $M_k$  is removed from the primary cache. In step 166, the  
22 movie  $M_i$  is moved from disk to cache.

1 In view of the above, it should be apparent that the video file server provides  
2 efficient management of access to movies ranging from very popular movies to  
3 unpopular movies by physically configuring sets of data movers for associated movie  
4 rankings, by reserving data mover local cache resources for the most popular movies,  
5 reserving a certain number of streams for popular movies, by negotiating with a client for  
6 selection of available movie titles during peak demand when resources are not available  
7 to start any freely-selected movie in disk storage, and by managing disk bandwidth and  
8 primary and local cache memory and bandwidth resources for popular and unpopular  
9 movies. The assignment of resources to movie rankings may remain the same while the  
10 rankings of the movies are adjusted, for example, during off-peak hours. A movie locked  
11 in primary cache and providing a source for servicing a number of video streams may be  
12 demoted from primary cache to disk in favor of servicing one or more streams of a  
13 higher-ranking movie (as described above with reference to FIG. 20).